

# Karan Gurazada

kgurazad@utexas.edu • [GitHub](#) • [LinkedIn](#)

## Education

**BS Computer Science '25** *The University of Texas at Austin—Turing Scholars Honors* August '21–  
*Classes*—Data Structures [Honors], Architecture [H], Operating Systems [H], Algorithms [H],  
Concurrency [H], Virtualization, Compilers, Programming for Performance, Prediction  
Mechanisms in Computer Architecture [Graduate], Distributed Systems [G], Advanced OS [G]

## Experience

**Software Engineer Intern** *CUDA Driver Team, NVIDIA [Santa Clara, CA]* May '24–  
Add new features, fix functionality bugs/performance regressions, accelerate workloads,  
engineer tests/testing systems across NVIDIA's GPU and Deep Learning software stack. September '24

- Fixed bug in **qemu** memory mapping for **vfio** devices using **linux kernel gdb**

**Software Engineer Intern** *Apple Silicon GPU Driver Team, Apple [Cupertino, CA]* May '23–  
Designed and implemented assembly-level debugger for GPU programs within the driver. August '23

- Built lightweight memory manager, worked across kernel-, user-mode codebases

**Undergraduate Researcher** *The University of Texas at Austin* June '22–  
Work closely with professors, senior PhD students on systems, chemistry research.

- *Linux CFS Testing*—Use **Racket** to generate corner cases for **multicore scheduling**
- *Starla*—Built optimized parallel DNN benchmark using **Xilinx HLS** for our novel **FPGA streaming** system, pending review in SOSP '24
- *PyAMFF*—Sped up **PyTorch** neural network training by **>60%** using in-memory data caching, **CUDA** profilers. Optimized small models with debug **TorchScript JIT** build
- *Kinetic Database*—Wrote optimized graph isomerism algorithm in **CUDA** with **Python** bindings on top of local **SQLite** or remote **MySQL** database

## Projects

**PopcornOS**—From-scratch operating system for **Intel i386** architecture written in **C++**

- Implemented **kernel multithreading** and **heap management**, ported **musl libc** to custom syscall API

**cp\_uring**—Optimized recursive directory copying on Linux using the **io\_uring** interface

- Achieved **>60%** speedup over GNU coreutils, used Linux **perf** and **qemu** tracing to profile heuristics

**Branch Golf**—Game in which the goal is to minimize runtime branches in **C/C++/Rust** code

- Modified **Clang/LLVM** to insert dynamic branch counting instructions into **Aarch64** ELF binaries
- Multithreaded backend in **Go**, frontend in **WebAssembly** compiled from **Rust**

**Hide and Seek**—Multiplayer mobile game played by >20 clients over large geographical area

- Built custom protocol with **UDP** and **FlatBuffers** for low-latency, fault-tolerant location streaming
- **iOS** frontend in **Swift** and **MapKit**, backend in **Rust** using **Tokio** and hosted on **DigitalOcean**

## Skills

**Languages**—C, C++, CSS, Elixir, Fortran, Go, HTML, Java, JavaScript, Perl, Python, R, Ruby, Rust, Shell, Swift, TypeScript, Verilog, x86/ARM/WebAssembly, Zig  
**Tools**—AWS, Boost, CUDA, Docker, GCC, GDB, Git, jQuery, ld, LLVM, Make, MPI, Ninja, Node.js, Nsight, perf, pip, PyTorch, qemu, SQLite, Tokio, Valgrind